HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200309618-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Carl Staelin et al. **Confirmation No.:** 6065

Application No.: 10/600,671 **Examiner:** Michael B. Holmes

Filing Date: June 20, 2003 **Group Art Unit:** 2121

Title: NEURAL NETWORK TRAINED WITH SPATIAL ERRORS

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on 9/21/2006.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☐ 1st Month $120 ☐ 2nd Month $450 ☐ 3rd Month $1020 ☐ 4th Month $1590

☐ The extension fee has already been filed in this application.

☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of $ 500. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

Respectfully submitted,

Carl Staelin et al.

By: /Hugh Gortler #33,890/

Hugh P. Gortler

Attorney/Agent for Applicant(s)

Reg No. : 33,890

Date : 11/21/2006

Telephone : (949) 454-0898

I hereby certify that this document is being transmitted to the Patent and Trademark Office via eletronic filing.

Date of Transmission: 11/21/2006

Patent
Docket No. 200309618-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

APPEAL NO. _________________

In re Application of:
Carl Staelin et al.

Serial No. 10/600,671
Filed: June 20, 2003

Confirmation No.6065
Group Art Unit: 2121
Examiner Michael B. Holmes

For: NEURAL NETWORK TRAINED WITH SPATIAL ERRORS

APPEAL BRIEF

Hugh P. Gortler, Esq.

Hewlett-Packard Company
Intellectual Property
Administration
P.O. Box 272400
Fort Collins, Colorado 80527-
2400

(949) 454-0898

INDEX

1. REAL PARTY IN INTEREST

The real party in interest is the assignee, Hewlett-Packard Development Company.

2. RELATED APPEALS AND INTERFERENCES

No appeals or interferences are known to have a bearing on the Board's decision in the pending appeal.

3. STATUS OF CLAIMS

Claims 1-37 are pending in this application.

Claims 8, 12 and 28 are objected to.

Claims 1-7, 9-11, 13-27 and 29-37 are rejected.

The rejections of claims 1-7, 9-11, 13-27 and 29-37 are being appealed.

4. STATUS OF AMENDMENTS

No amendment has been filed subsequent to the final office action, dated July 21, 2006. The claims are listed in appendix A.

5. SUMMARY OF CLAIMED SUBJECT MATTER

Image upscaling typically involves magnifying an entire image or a portion of an image. For example, an image upscaled by a factor of two may have a 2x2 block of pixels corresponding to each pixel in the original image. Pixel values in each 2x2 block of the upscaled image are predicted from pixel values in the original image.

Neural networks can be used to perform image upscaling. As described in paragraph 19 of the specification, a neural network is defined by its nodes, connections, and connection weights. A weight vector is the vector of connection weights between each pair of connected nodes in the neural network. An example of a neural network is illustrated in Figure 5 and described in paragraph 51 of the specification. The exemplary neural network 512 includes input, hidden and output nodes 512-516.

A neural network can be trained to upscale an image. A high resolution image may be downsampled (e.g., by pixel averaging) to produce a corresponding low resolution image. The high resolution image is referred to as a target image. The low resolution image is inputted to the neural network, which produces an upscaled image. Training involves optimizing these weight values so as to reduce the error between the upscaled image and the high resolution image.

Base claim 1 recites a method of training a neural network with input data. The method comprises using the neural network to rescale the input data, determining errors for the rescaled data, and using neighborhoods of the errors to adjust connection weights of the neural network.

An example of training a neural network for image upscaling is illustrated in Figure 3b and described in paragraphs 24-35 of the specification. The neural network is used to upscale the input image (block 314 of Figure 3b and paragraph 26 of the specification). Errors for the upscaled (output) image are determined (block 316 of Figure 3b and paragraph 27 of the specification). For example, an error image may be formed by subtracting the upscaled image from the target image.

Neighborhoods of the errors are used to adjust the connection weights (blocks 318-322 of Figure 3b and paragraphs 28-33 of the specification). An exemplary neighborhood of image pixels is illustrated in Figure 3a. The pixel being processed is the center pixel z_{11}.

Paragraph 28 provides an example of computing derivatives of the errors with respect to the upscaled image (block 318). The derivative for a predicted pixel in the upscaled image is a function of differences between predicted values in a spatial neighborhood of the upscaled image and the corresponding pixel values in the target image. A simple function is described, in which the derivative is the sum of partial derivatives of the neighboring pixels in a spatial neighborhood.

According to paragraph 31, once the full derivatives for the pixels in the rescaled image have been generated, back-propagation is performed to compute error gradients (block 320). According to paragraph 31, the error gradients are used to adjust the node weights to reduce the network errors (block 322).

An example of non-gradient based training is illustrated in Figure 8 and described in paragraph 69. The non-gradient based training also makes use of neighborhoods of errors.

Base claim 19 recites a method of using input data and target data to train a neural network. The method comprises using the neural network to generate predicted values from the input data, determining errors for the predicted values, and back-propagating the errors through the neural network. The error for each predicted value is a function of differences between predicted values in a spatial neighborhood and the corresponding values in the target data.

An example of training a neural network for image upscaling is illustrated in Figure 3b and described in paragraphs 24-35 of the specification. The neural network is used to generate predicted values as illustrated by block 314 of Figure 3b and described in paragraph 26 of the specification. The errors for the predicted values are determined as illustrated by blocks 316-318 of Figure 3b and described in paragraphs 28-30. Back propagation is illustrated by block 320 of Figure 3b and described in paragraphs 31-32.

Base claim 20 recites apparatus for training a neural network on input data. The apparatus comprises means for using the neural network to rescale the input data, means for determining errors for the rescaled data, and means for using neighborhoods of the errors to adjust the connection weights.

An exemplary apparatus is illustrated in Figure 7 and described in paragraphs 80-83 of the specification. A processor 712 and memory 714 provide the means for using the neural network to rescale the input data, determining errors for the rescaled data, and using neighborhoods of the errors to adjust the connection weights. An example of rescaling the input data, determining the errors, and using the errors to adjust connection

weights is described in paragraphs 26-33 and illustrated by blocks 314-322 of Figure 3b.

Base claim 21 recites apparatus for training a neural network on input data. The apparatus comprises a processor programmed to use the neural network to rescale the input data, determine errors for the rescaled data, and use neighborhoods of the errors to adjust connection weights of the neural network.

An exemplary apparatus is illustrated in Figure 7 and described in paragraphs 80-83 of the specification. The processor is referenced by numeral 712. An example of rescaling the input data, determining the errors, and using the errors to adjust connection weights is described in paragraphs 26-33 and illustrated by blocks 314-322 of Figure 3b.

Base claim 34 recites apparatus for rescaling a color image. The apparatus comprises means for rescaling the input image by pixel replication, and a neural network that has been trained to rescale a luminance channel of the color image. The neural network produces a rescaled luminance image. The apparatus further comprises means for using the rescaled luminance image and the pixel-replicated image to generate a rescaled color image.

An exemplary apparatus is illustrated in Figure 7 and described in paragraphs 80-83 of the specification. A processor 712 and memory 714 provide the neural network, the means for rescaling the input image, and the means for generating the rescaled color image. An example of rescaling the input data, determining the errors, and using the errors to adjust connection weights for a single color channel is described in paragraphs 26-33 and illustrated by blocks 314-322 of Figure 3b.

Examples of extending the rescaling to multiple color channels are illustrated in Figure 6a-6c and described in paragraphs 70-75.

Base claim 37 recites an article for causing a processor to use input data to adjust connection weights of a neural network. The article comprises computer memory and data encoded in the computer memory. The data causes the processor to use the neural network to rescale the input data, determine errors for the rescaled data, and use neighborhoods of the errors to adjust the connection weights of the neural network.

An example of the article is illustrated in Figure 7 and described in paragraphs 80-81 of the specification. The article includes memory 714 encoded with a program 720 and a neural network 716. An example of rescaling the input data, determining the errors, and using the errors to adjust connection weights is described in paragraphs 26-33 and illustrated by blocks 314-322 of Figure 3b.

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

a. Base claims 1, 19-21, and 37 are rejected under 35 USC §102(b) as being anticipated by Skeirik U.S. Patent No. 5,826,249.

b. Base claim 34 is rejected under 35 USC §102(b) as being anticipated by Skeirik U.S. Patent No. 5,826,249.

7. ARGUMENTS

I

REJECTION OF BASE CLAIMS 1, 19-21 AND 37 UNDER 35 U.S.C. §102 AS BEING ANTCIPATED BY SKERIK

Skeirik describes a neural network for the monitoring and control of manufacturing processes (col. 1, lines 26-29). A representative embodiment of training such a neural network is disclosed at col. 20, lines 46+. During training of the neural network, training input data is retrieved and scaled using high and low limit values (col. 23, lines 23-26), and then fed to the neural network, which predicts output data (col. 23, lines 27-28). The output data is then de-scaled (col. 23, lines 33-35). Error data is computed from the de-scaled output data and the unscaled input data (col. 23, lines 35-44). The neural network is retrained using the error data (col. 23, lines 45-49).

Skeirik does not teach or suggest using a neural net to perform scaling of input data. Scaling and de-scaling are performed before and

after the neural network is used. Although Skeirik doesn't explain why the scaling is performed with high and low limit values, such scaling is probably performed in order to accommodate the architectural restrictions of the neural network.

Skeirik does not teach or suggest determining errors for the rescaled training data. Skeirik determines errors at the original scale by comparing a de-scaled output image to the non-scaled training data.

Skeirik does not teach or suggest using neighborhoods of errors to adjust network connection weights. Skeirik does not describe how the neural network is retrained, other than saying back propagation (Fig. 34 and col. 12, lines 4+) or another method can be used (col. 23, lines 48-49). Col. 12, lines 31+ and block 3404 of Figure 34 do not indicate how errors are computed. Skeirik is silent about neighborhoods of errors. He does not teach or suggest that output errors are correlated with other output errors, nor does he teach or suggest that these correlations can be exploited.

Therefore, the examiner has no basis to allege that Skeirik uses neighborhoods of errors to adjust network connection weights. The examiner's allegation regarding neighborhoods of errors is unsubstantiated. The examiner's unsubstantiated allegation was challenged in the previous response, and it is challenged here.

Skeirik does not teach or suggest (1) using a neural network to rescale input data, (2) determining errors for rescaled data, or (3) using neighborhoods of errors to adjust network connection weights. Therefore, claim 1 and its dependent claims 2-18 should be allowed over Skeirik.

Base claims 19-21, and 37 and their dependent claims should be allowed for the same reasons.

Rather than provide evidence in the prior art of (1) determining errors for rescaled data and (2) using neighborhoods of errors to adjust network connection weights, the examiner attempts to render the claims unpatentable by applying unreasonably overbroad interpretations to the claim language. In paragraph 12 of the final office action, the examiner uses Webster's dictionary to define the term rescaled as "to plan, establish or formulate on a new and usu. smaller scale." This definition of rescale is irrelevant, since Webster's doesn't define it in the context of neural networks. Moreover, the definition doesn't address the issues of whether Skeirik's neural network performs scaling (Skeirik doesn't), and whether errors are determined for rescaled data (Skeirik computes errors at the same scale as the input data).

Also in paragraph 12 of the final office action, the examiner interprets neighborhood as "training data." However, the examiner does not explain how he arrives at this interpretation, nor does he offer evidence to support his interpretation. The examiner's interpretation is arbitrary and capricious. He does not follow Phillips vs. AWH Corp. , which

states an examiner is supposed to provide a reasonable interpretation of the claims in view of the specification, and determine whether the claims read on the prior art. Phillips v. AWH Corp., 03-1269, -1286, p. 16 (Fed.Cir. 2005) citing In re Am.Acad.of.Sci.Tech. Ctr., 367 F.3d 1359 (Fed.Cir. 2004).

Moreover, the examiner's definition of neighborhood is inconsistent with the claim language. Claim 1 recites "using neighborhoods of the errors [of the rescaled data] to adjust the connection weights." Substituting the examiner's definition for neighborhood, claim 1 would recite *using training data of the rescaled data errors*" to adjust connection weights. The examiner's interpretation of neighborhood does not make sense.

II
REJECTION OF BASE CLAIM 34 UNDER 35 U.S.C. §102 AS BEING ANTCIPATED BY SKERIK

Base claim 34 should be allowed for the same reasons that base claim 1 should be allowed. Argument I is incorporated by reference.

Base claim 34 and its dependent claims 35-36 should be allowed for the additional reason that Skeirik doesn't teach or suggest a neural network for rescaling a color image. Skeirik describes a neural network for the monitoring and control of manufacturing processes (col. 1, lines 26-29).

The rescaling of claim 34 involves taking an input pattern with elements in a certain spatial (distance) relationship (the color image) and producing an output pattern of a different dimensionality that preserves key properties of the input distance relationship (the rescaled image). Skeirik's method involves taking time-dependent input data and training a neural network to be sensitive to the time-series nature of the input data.

Moreover, Skeirik is not analogous art with respect to claim 34. MPEP 2141.01(a) states "In order to rely on a reference as a basis for rejection of an applicant's invention, the reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the inventor was concerned." Skeirik's field involves the monitoring and control of manufacturing processes (col. 1,

lines 26-29), whereas the field of claim 34 involves rescaling a color image. Skeirik is faced with the problems of determining set points for manufacturing processes, and maintaining process conditions at those set points during manufacture (col. 2, lines 42+ in general; and col. 3, lines 38-52 and col. 6, lines 7-17 in particular). Image resizing is not concerned with such problems.

For the reasons above, the rejections of claims 1-7, 9-11, 13-27 and 29-37 should be withdrawn. The Honorable Board of Patent Appeals and Interferences is respectfully requested to reverse these rejections.

Respectfully submitted,

/Hugh Gortler #33,890/
Hugh P. Gortler, Esq.
Registration No. 33, 890

Hewlett-Packard Company
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

(949) 454-0898

Date: November 21, 2006

8. CLAIMS APPENDIX

1. (Original) A method of training a neural network with input data, the neural network including a plurality of connection weights, the method comprising:

using the neural network to rescale the input data;

determining errors for the rescaled data; and

using neighborhoods of the errors to adjust the connection weights.

2. (Original) The method of claim 1, wherein the input data represents a set of images, and wherein the neighborhoods are spatial error neighborhoods.

3. (Original) The method of claim 1, wherein the error neighborhoods are used with a non-gradient algorithm to adjust the connection weights.

4. (Original) The method of claim 1, wherein the error neighborhoods are used to generate derivatives of total error with respect to a neighborhood of errors; wherein gradients are computed from the derivatives; and wherein the gradients are used to adjust the connection weights.

5. (Original) The method of claim 4, wherein each derivative is computed as the sum of the partial derivatives of the errors in an error neighborhood.

6. (Original) The method of claim 4, wherein each derivative of total error with respect to a neighborhood of errors is proportional to a product of a penalty matrix and an error vector, the error vector describing the neighborhood of errors, the penalty matrix punishing any spatially correlated errors.

7. (Original) The method of claim 6, wherein the penalty matrix is positive definite, and includes weights that penalize undesirable patterns of errors.

8. (Original) The method of claim 6, wherein the penalty matrix is based on use of a pattern detector that detects the spatially correlated errors.

9. (Original) The method of claim 1, wherein determining the errors includes forming an error image from the rescaled data, identifying patterns in the error image, and punishing the spatially correlated errors in the error image.

10. (Original) The method of claim 1, wherein input and output data of the neural network are coded to improve the neural network accuracy.

11. (Original) The method of claim 1, wherein the errors are a combination of SSE and spatial errors.

12. (Original) The method of claim 11, wherein SSE is applied to crisp edges and spatial errors are applied to blurred edges.

13. (Original) A method of upscaling an input image, the method comprising using the neural network trained according to claim 1.

14. (Original) The method of claim 13, wherein the input and upscaled images are color images; wherein the input image is upscaled by pixel replication; a luminance channel of the input image is upscaled by the neural network; and the upscaled luminance channel and the pixel-replicated image are used to generate the upscaled color image.

15. (Original) The method of claim 14, wherein the using the upscaled luminance channel and the pixel-replicated image include adding deltas to pixels in the pixel-replicated image, each delta computed as the difference between the corresponding luminance value in the upscaled luminance channel and the corresponding luminance value in the input luminance channel.

16. (Original) The method of claim 15, wherein using the upscaled luminance channel and the pixel-replicated image further includes gamut mapping the upscaled image.

17. (Original) An article comprising computer memory encoded with data upscaled by the neural network trained according to the method of claim 1.

18. (Original) Apparatus comprising a processor programmed with a neural network, the network trained according to the method of claim 1.

19. (Original) A method of using input data and target data to train a neural network, during training, the method comprising:

using the neural network to generate predicted values from the input data;

determining errors for the predicted values, the error for each predicted value a function of differences between predicted values in a spatial neighborhood and the corresponding values in the target data; and

back-propagating the errors through the neural network.

20. (Original) Apparatus for training a neural network on input data, the apparatus comprising:

means for using the neural network to rescale the input data;

means for determining errors for the rescaled data; and

means for using neighborhoods of the errors to adjust the connection weights.

21. (Original) Apparatus for training a neural network on input data, the neural network having a plurality of connection weights, the apparatus comprising a processor programmed to use the neural network to rescale the input data; determine errors for the rescaled data; and use neighborhoods of the errors to adjust the connection weights of the neural network.

22. (Original) The apparatus of claim 21, wherein the input data represents images, and wherein the neighborhoods are spatial error neighborhoods.

23. (Original) The apparatus of claim 21, wherein the processor is programmed to use the error neighborhoods and a non-gradient algorithm to adjust the connection weights.

24. (Original) The apparatus of claim 21, wherein the error neighborhoods are used to generate derivatives of total error with respect to a neighborhood of errors; wherein gradients are computed from the derivatives; and wherein the gradients are used to adjust the connection weights.

25. (Original) The apparatus of claim 24, wherein each derivative is computed as the sum of the partial derivatives of the errors in an error neighborhood.

26. (Original) The apparatus of claim 24, wherein each derivative of total error with respect to a neighborhood of errors is proportional to a product of a penalty matrix and an error vector, the error vector describing the neighborhood of errors, the penalty matrix punishing any spatially correlated errors.

27. (Original) The apparatus of claim 26, wherein the penalty matrix is positive definite, and includes weights that penalize undesirable patterns of errors.

28. (Original) The apparatus of claim 26, wherein the penalty matrix is based on use of a pattern detector that detects the spatially correlated errors.

29. (Original) The apparatus of claim 21, wherein determining the errors includes forming an error image from the rescaled data, identifying patterns in the error image, and punishing the spatially correlated errors in the error image.

30. (Original) The apparatus of claim 21, wherein the processor is programmed to code input and output data of the neural network to improve the neural network accuracy.

31. (Original) The apparatus of claim 21, wherein the errors are a combination of SSE and spatial errors.

32. (Original) The apparatus of claim 21, wherein the input and upscaled images are color images; wherein the processor is programmed to upscale the input image by pixel replication, use the neural network to upscale a luminance channel of the input image; and generate the upscaled color image from the upscaled luminance channel and the pixel-replicated image.

33. (Original) The apparatus of claim 32, wherein the processor is further programmed to perform gamut mapping of the upscaled image.

34. (Original) Apparatus for rescaling a color image, the apparatus comprising:

means for rescaling the input image by pixel replication;

a neural network that has been trained to rescale a luminance channel of the color image, the neural network for producing a rescaled luminance image; and

means for using the rescaled luminance image and the pixel-replicated image to generate a rescaled color image.

35. (Original) The apparatus of claim 34, wherein the use of the rescaled luminance image and the pixel-replicated image includes adding deltas to pixels in the pixel-replicated image, each delta computed as the difference between the corresponding luminance value in the rescaled luminance image and the corresponding luminance value in the input luminance channel.

36. (Original) The apparatus of claim 32, further comprising means for gamut mapping the rescaled color image.

37. (Original) An article for causing a processor to use input data to adjust connection weights of a neural network, the article comprising:

computer memory:

data encoded in the computer memory, the data causing the processor to use the neural network to rescale the input data; determine errors for the rescaled data; and use neighborhoods of the errors to adjust the connection weights of the neural network.

9. <u>EVIDENCE APPENDIX</u>

None

10. <u>RELATED PROCEEDINGS APPENDIX</u>

None